VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra aplikované matematiky

Generátory pseudonáhodných čísel

Random Number Generators

2014 Lukáš Mihula

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra aplikované matematiky

Zadání bakalářské práce

Student: **Lukáš Mihula**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 1103R031 Výpočetní matematika

Téma: Generátory pseudonáhodných čísel
Random Number Generators

Zásady pro vypracování:

Po popisu základních metod generování pseudonáhodných čísel bude analyzována kvalita vybraných generátorů, včetně numerických experimentů a statistických testů.
After describing the basic methods of generating pseudo-random numbers will be analyzed by the quality of the selected generators, including numerical experiments and statistical tests.

Seznam doporučené odborné literatury:

Antoch, J.: Jak pomocí simulací dokázat nemožné, Informační Bulletin České Statistické Společnosti, ročník 9., č. 1, 1998
Máša, P.: Zajímavý generátor náhodných čísel, Informační Bulletin České Statistické Společnosti, ročník 14., č. 4, 2003

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Mgr. Bohumil Krajc, Ph.D.**

Datum zadání: 01.09.2013
Datum odevzdání: 07.05.2014

VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA OSTRAVA
Fakulta elektrotechniky a informatiky
OSTRAVA

doc. RNDr. Jiří Bouchala, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2014 .....

Abstrakt

Generátory pseudonáhodných čísel mají v dnešní době uplatnění v několika různých oborech. Od použití v hracích automatech přes kryptografii až po simulaci nukleárních reakcí. Předponou pseudo odlišujeme deterministický generátor od tzv. pravého generátoru náhodných čísel, který využívá fyzikální a hardwarové metody generování. Generování pomocí pravých generátorů náhodných čísel je pomalé, drahé nebo těžko kontrolovatelné, a proto se používají generátory softwarové, tedy pseudonáhodné. V této práci popisuji několik často používaných pseudonáhodných generátorů, které následně testuji řadou statistických testů pro ověření jejich kvality.

Klíčová slova: generátor pseudonáhodných čísel, NIST, statistické testy, náhodná veličina, náhodný vektor, zamítací metoda, metoda inverzní transformace

Abstract

There are several different fields of study in which pseudorandom number generators are used these days. They are being deployed from slot machines over cryptography to simulations of nuclear reaction. By the prefix “pseudo” we differentiate deterministic generator from so called true generator of random numbers which uses physical and hardware methods of generation. Generation by true generators of random numbers is slow, expensive and hard to control, thus software generators - pseudorandom generators are used. In this work I describe several pseudorandom generators which are often used and which I test afterwards by statistical testing for checking their quality.

Keywords: random number generator, NIST, statistical tests, random variable, random vector, accept-reject algorithm, inverse transform method

Seznam použitých zkratek a symbolů

BBS	–	Blum Blum Shub
ICG	–	Inversive Congruential Generator
LCG	–	Linear Congruential Generator
LFG	–	Lagged Fibonacci Generator
LSFR	–	Linear Feedback Shift Register
MT	–	Mersenne Twister
NIST	–	National Institute of Standards and Technology
PRNG	–	Pseudorandom Number Generator
TG	–	Tausworthe Generator
χ^2	–	Chí-kvadrát

Obsah

Seznam tabulek

Seznam obrázků

1 Úvod

Generátory náhodných čísel jsou nedílnou součástí teorie simulací v oboru informatiky a inženýrství. Na základě druhu metody generování existují dva typy generátorů - pseudonáhodné a pravé náhodné generátory. Pravé náhodné generátory využívají ke generování čísel fyzikální jevy, např. atmosférický šum, kosmické záření nebo radioaktivní rozpad. Pseudonáhodné generátory generují (pseudo)náhodná čísla pomocí deterministického algoritmu, kde veškerá náhodnost výstupu je závislá na náhodnosti vstupních hodnot. V této práci se budu zabývat generátory pseudonáhodných čísel (PRNG).

Na generátory pseudonáhodných čísel jsou díky jejich hojnému použití kladeny vysoké nároky. I když pravé náhodné generátory budou vždy generovat „náhodnější" čísla, pseudonáhodné generátory budou díky své rychlosti a jednoduchosti implementace používanější. Na základě oblasti svého použití musí generátor dosahovat určité kvality. U některých aplikací „statistická" kvalita generátoru nemusí hrát velkou roli (kryptografie), ale u řady problémů, které využívají generování náhodných čísel, jako techniky simulace Monte Carlo, bude potřeba „náhodnosti" generátoru zásadní. Generátory obvykle nejprve generují čísla s uniformním rozdělením na intervalu $(0, 1)$, neuniformní rozdělení se potom vytváří pomocí transformace. Cílem pseudonáhodných generátorů je generovat sekvence čísel, které jsou v praxi nerozpoznatelné od pravých náhodných hodnot. Ověření kvality vygenerovaných dat bývá často založeno na statistických testech. Tyto testy ověřují náhodnost hledáním korelací a vyšetřováním dalších důležitých vlastností.

V první části této práce se věnuji jednotlivým druhům pseudonáhodných generátorů a seznámení čtenáře s jejich vlastnostmi a použitím. V druhé části práce se věnuji testování generátorů používaných ve známých aplikacích jako MATLAB, Maple, ad. Na základě těchto testů jsou pak zhodnoceny jednotlivé generátory a použité algoritmy.

2 Úvodní poznámky

Tato kapitola slouží zejména ke sjednocení zápisu a upřesnění některých pojmů ze statistiky a teorie pravděpodobnosti. Předpokládá se, že čtenář má základní znalost těchto odvětví matematiky. Více o této problematice v [8, 9].

2.1 Základní pojmy

Náhodná veličina

Nechť je dán pravděpodobnostní prostor (Ω, S, P). Náhodná veličina je zobrazení $X : \Omega \to \mathbb{R}$ takové, že pro každé $x \in \mathbb{R}$ je množina

$$\{\omega \in \Omega : X(\omega) \leq x\}$$

prvkem S, tedy náhodným jevem.

Distribuční funkce

Funkce $F : \mathbb{R} \to \mathbb{R}$ definovaná vztahem

$$F(x) = P(X \leq x)$$

se nazývá distribuční funkcí náhodné veličiny X.

Hustota pravděpodobnosti

Hustota pravděpodobnosti f spojité náhodné veličiny s distribuční funkcí F je reálná nezáporná funkce taková, že

$$F(x) = \int_{-\infty}^{x} f(t)\,\mathrm{d}t, \quad \text{pro} -\infty < x < \infty.$$

Obdoba hustoty pravděpodobnosti pro diskrétní veličinu je tzv. pravděpodobnostní funkce, kterou budeme značit P.

Střední hodnota

Střední hodnota ($E(x)$ nebo μ) je důležitým parametrem rozdělení náhodné veličiny. Pro diskrétní náhodnou veličinu je dána vztahem

$$\mu = \sum_{(i)} x_i \cdot P(x_i).$$

Pro spojitou náhodnou veličinu je dána předpisem

$$\mu = \int_{-\infty}^{\infty} x \cdot f(x)\,\mathrm{d}x.$$

Rozptyl

Rozptyl ($D(x)$ nebo σ^2) je dalším důležitým parametrem rozdělení náhodné veličiny. Pro diskrétní náhodnou veličinu je určen vztahem

$$\sigma^2 = \sum_{(i)} (x_i - E(x))^2 \cdot P(x_i).$$

Pro spojitou náhodnou veličinu je dán vzorcem

$$\sigma^2 = \int_{-\infty}^{\infty} (x - E(x))^2 \cdot f(x)\,\mathrm{d}x.$$

2.2 Některá rozdělení pravděpodobnosti

Uniformní rozdělení

Náhodna veličina X s uniformním rozdělením má konstantní hustotu pravděpodobnosti na intervalu (a, b) a nulovou mimo něj. Zapisuje se ve tvaru $U \to U(a, b)$ nebo $U(a, b)$. Hustota pravděpodobnosti, distribuční funkce, střední hodnota a rozptyl uniformního rozdělení jsou:

$$f(x) = \begin{cases} \frac{1}{b-a}, & x \in (a, b) \\ 0, & x \notin (a, b) \end{cases}, F(x) = \begin{cases} 0, & x \in (-\infty, a\rangle \\ \frac{1}{b-a}, & x \in (a, b\rangle \\ 1, & x \in (b, \infty) \end{cases},$$

$$\mu = \frac{a+b}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12}.$$

Poznámka 2.1 (Diskrétní uniformní rozdělení) Náhodná veličina s diskrétním uniformním rozdělením může nabývat pouze konečně mnoha hodnot $n \in \mathbb{N}$. Všechny tyto hodnoty mají stejnou pravděpodobnost výskytu $\frac{1}{n}$ a předpokládá se, že vzdálenost mezi jednotlivými hodnotami je stejná.

Alternativní (Bernoulliho) rozdělení

Alternativní rozdělení náhodné veličiny X udává počet úspěchů náhodného jevu v jednom pokusu. Zapisuje se ve tvaru $X \to \mathrm{Ber}(p)$, kde p tzv. je pravděpodobnost úspěchu pokusu. Pravděpodobnostní funkce, střední hodnota a rozptyl alternativního rozdělení jsou:

$$P(x) = \begin{cases} p, & x = 1 \\ (1-p), & x = 0 \end{cases}, \quad \mu = p, \quad \sigma^2 = p(p-1).$$

Binomické rozdělení

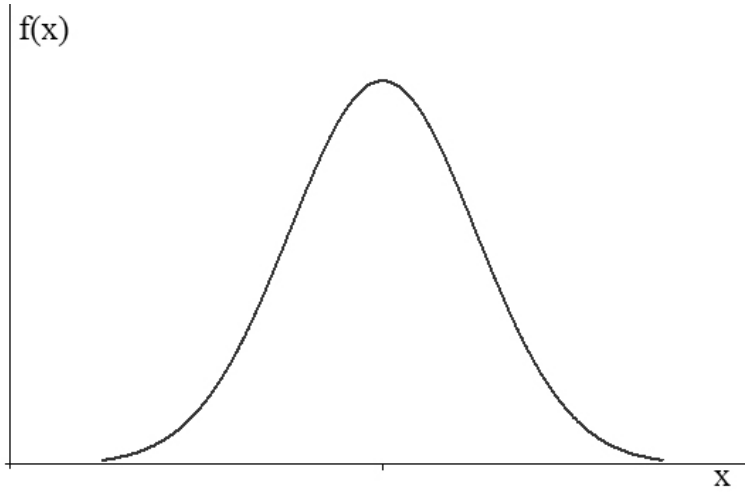
Binomické rozdělení náhodné veličiny X udává počet úspěchů náhodného jevu v n Bernoulliho pokusech. Zapisuje se ve tvaru $X \to \mathrm{Bi}(n, p)$, kde n je celkový počet pokusů a p pravděpodobnost úspěchu v každém z pokusů. Součtem n nezávislých alternativních rozdělení lze získat náhodnou veličinu s binomickým rozdělením. Pravděpodobnostní funkce, střední hodnota a rozptyl binomického rozdělení jsou:

$$P(x) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x}, & x \in \{0, 1, ..., n\} \\ 0, & x \notin \{0, 1, ..., n\} \end{cases}, \quad \mu = np, \quad \sigma^2 = np(p-1).$$

Normální rozdělení

Normální rozdělení je nejpoužívanějším pravděpodobnostním rozdělením, které modeluje chování řady náhodných jevů v různých odvětvích. Zapisuje se ve tvaru $X \to N(\mu, \sigma^2)$, kde μ je střední hodnota a σ^2 je rozptyl. Řada pravděpodobnostních rozděleních lze za určitých podmínek tímto rozdělením aproximovat. Hustota pravděpodobnosti normálního rozdělení je:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(\frac{x-\mu}{\sqrt{2}\sigma})^2}, \quad \text{pro } -\infty < x < \infty.$$



Obrázek 1: Hustota pravděpodobnosti normálního rozdělení

X^2 rozdělení

Chí kvadrát rozdělení je rozdělením náhodné veličiny, která vznikne jako součet čtvreců nezávislých náhodných veličin s normovaným normálním rozdělením. Chí kvadrát rozdělení se označuje jako $X \to \chi^2_n$ kde tzv. je počet stupnů volnosti n označuje počet sčítaných náhodných veličin. Hustota pravděpodobnosti chí kvadrát rozdělení je:

$$f(x) = \begin{cases} \frac{1}{\Gamma(\frac{n}{2})2^{\frac{n}{2}}} e^{-\frac{x}{2}} x^{\frac{n}{2}-1} & \text{pro } x > 0 \\ 0, & \text{pro } x \leq 0 \end{cases}, \quad \mu = n, \quad \sigma^2 = 2n.$$

kde Γ je gama funkce definová předpisem:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} \mathrm{dt}. \tag{1}$$

2.3 Náhodný vektor

Nechť je dán pravděpodobnostní prostor (Ω, φ, P). Budiž X, Y náhodné veličiny na tomto prostoru. Pak dvojici (X, Y) nazveme **náhodným vektorem**. Funkci $F : \mathbb{R}^2 \to \langle 0, 1 \rangle$ definovanou předpisem

$$F_{XY}(x, y) = P(X < x, Y < y)$$

nazýváme sdruženou distribuční funkcí vektoru (X, Y). Nechť existuje nezáporná funkce $f_{XY} : \mathbb{R}^2 \to \mathbb{R}$ taková, že pro každé $(x, y) \in \mathbb{R}^2$ platí

$$F_{XY}(x, y) = \iint\limits_{(-\infty,x\rangle\times(-\infty,y\rangle} f_{XY}(r, s)\mathrm{d}r\mathrm{d}s.$$

Pak řekneme, že (X, Y) je spojitý náhodný vektor a funkci f_{XY} říkáme **sdružená hustota** vektoru (X, Y).

Věta 2.1 *Nechť $M \subset \mathbb{R}^2$ je meřitelná množina. Pak*

$$P((X, Y) \in M) = \iint\limits_{M} f_{XY}(x, y)\mathrm{d}x\mathrm{d}y.$$

Nechť (X, Y) je náhodný vektor. Pak funkcím

$$F_X(x) = P(X < x),$$
$$F_Y(y) = P(Y < y),$$

říkáme **marginální distribuční funkce** náhodného vektoru (X, Y). Buď (X, Y) spojitý náhodný vektor. Nechť pro každé $(x, y) \in \mathbb{R}^2$ platí:

$$F_X(x) = \int\limits_{-\infty}^{x} f_X(x)\mathrm{d}x,$$
$$F_Y(y) = \int\limits_{-\infty}^{y} f_Y(y)\mathrm{d}y.$$

Pak funkcím f_X, f_Y říkáme **marginální hustoty**.

Věta 2.2 *Je-li f_{XY} sdružená hustota spojitého náhodného vektoru (X, Y), pak pro marginální hustoty platí:*

$$f_X(x) = \int\limits_{-\infty}^{\infty} f_{XY}(x, y)\mathrm{d}y,$$
$$f_Y(y) = \int\limits_{-\infty}^{\infty} f_{XY}(x, y)\mathrm{d}x.$$

Řekněme, že náhodné veličiny X, Y na stejném pravděpodobnostním prostoru (Ω, S, P) jsou **nezávislé**, pokud pro libovolné borelovské množiny $A \subset \mathbb{R}, B \subset \mathbb{R}$ jsou jevy

$$\{\omega \in \Omega : X(\omega) \in A\}, \quad \{\omega \in \Omega : X(\omega) \in B\}$$

nezávislé.

Věta 2.3 *Nechť X, Y jsou nezávislé náhodné veličiny. Pak platí:*

$$F_{XY}(x, y) = F_X(x) \cdot F_Y(x).$$

Je-li navíc (X, Y) spojitý náhodný vektor, pak

$$f_{XY}(x, y) = f_X(x) \cdot f_Y(x).$$

Nechť (X, Y) je spojitý náhodný vektor. Pak pro $y \in \mathbb{R}$ takové, že $f_Y(y) \neq 0$ definujeme **podmíněnou pravděpodobnostní hustotu** veličiny X (za podmínky $Y = y$) předpisem

$$f_{X|Y}(x|Y = y) = \frac{f_{XY}(x, y)}{f_Y(y)}.$$

Věta 2.4 *Budiž $A \subset \mathbb{R}$ měřitelná množina. Potom*

$$P(x \in A \,|Y = y) = \int_A f_{X|Y}(x|Y = y)\mathrm{d}x.$$

Poznámka 2.2 Podobně definujeme podmíněnou hustotu

$$f_{Y|X}(y|X = x) = \frac{f_{XY}(x, y)}{f_X(x)}.$$

Věta 2.5 *Nechť X, Y jsou nezávislé náhodné veličiny. Potom*

$$f_{X|Y}(x|Y = y) = f_X(x),$$
$$f_{Y|X}(y|X = x) = f_Y(y),$$

za předpokladu, že funkce na obou stranách rovností jsou definovány.

2.4 Testování hypotéz

Poznámka 2.3 Statistická hypotéza je určité tvrzení o rozdělení pozorované veličiny. Toto tvrzení je založeno na základě předchozích zkušeností, na analýze dosavadních znalostí nebo na pouhém odhadu.

Při testování hypotéz se na statistickou hypotézu nahlíží nejprve jako na pravdivou a cílem statistického testu je ji vyvrátit (obdoba presumpce neviny). Pokud se to nepodaří, tak stoupne věrohodnost této hypotézy. Testování hypotéz je technicky pojato jako proces, ve kterém proti sobě stojí dvě tvrzení - nulová a alternativní hypotéza.

Nulová hypotéza H_0 je hypotéza, kterou testujeme. Označuje tvrzení, které je bráno jako předpoklad při testování.

Alternativní hypotéza H_A představuje hypotézu, která (vhodným způsobem) popírá tvrzení nulové hypotézy.

Na základě testu statistické hypotézy můžeme dojít ke dvěma rozhodnutím:

1. Zamítáme H_0 ve prospěch hypotézy H_A.
2. Nezamítáme H_0.

Důležitým parametrem testování hypotéz je tzv. hladina významnosti α. Hladina významnosti je pravděpodobnost, že se zamítné nulová hypotéza, ačkoliv platí. Zadáním hladiny významnosti α se obor hodnot testovaného parametru se dělí na obor přijetí V a kritický obor W. Hranice mezi těmito obory se označuje jako kritická hodnota testu. Padne-li pozorovaná hodnota testového parametru do kritického oboru W, zamítáme nulovou hypotézu. V opačném případě nulovou hypotézu nezamítáme.

Poznámka 2.4 Testová statistika $T(x)$ je funkce výběru, která vyjadřuje sílu platnosti nulové hypotézy vůči hypotéze alternativní. Je-li testová statistika v kritickém oboru, zamítámé nulovou hypotézu. Je-li testová statistika v oboru přijetí, nulovou hypotézu nezamítáme.

Jiný přístup k testování hypotéz je založen na tzv. stanovení p-hodnoty. p-hodnota je taková hodnota testové statistiky, která je hraničním bodem kritického oboru odpovídajícího hladině významnosti p. Jinak řečeno p-hodnota vyjadřuje, jaká je minimální hladina významnosti α, na níž bychom mohli (hraničně) zamítnout nulovou hypotézu.

3 Generátory pseudonáhodných čísel s uniformní rozdělením pravděpodobnosti

Vytváření vzorků náhodné veličiny s předepsaným rozdělením pravděpodobnosti je zásadní pro mnohé simulace a statistické výpočty. Tyto vzorky mohou být získány z náhodných fyzikálních jevů. I když se náhodný fyzikální jev zdá jako vhodná volba pro generování velkého množství dat, jeho použití je díky náročnosti a rychlosti generování často nereálné. Proto se v praxi používají ke generování náhodných čísel deterministické algoritmy. V této části popíšeme metody vytváření vzorků, které odpovídají uniformnímu rozdělení.

Obvykle algoritmus vychází z počáteční hodnoty $x_0 \in \Omega$ v prostoru možných hodnot Ω, na kterou se aplikuje pevně stanovené rozbrazení $D : \Omega \to \Omega$, čímž vznikne další hodnota $x_1 = D(x_0)$. Následující hodnoty se potom generují (deterministicky) stejným způsobem:

$$x_n = D(x_{n-1}) = D^n(x_0). \tag{2}$$

Proto se tyto vzorky označují jako pseudonáhodné. Při vhodné volbě zobrazení D a počáteční hodnoty x_0 je získaná posloupnost pomocí běžných testů nerozlišitelná od posloupnosti náhodných vzorků. Speciálně, pokud například $\Omega = (0, 1)$, pak by vygenerovaná posloupnost měla splňovat několik základních podmínek - projít testy na uniformitu rozdělení, mít velkou periodu opakování[1], nesouvztažnost atd.

Existuje mnoho generátorů pseudonáhodných čísel, v této kapitole se seznámíme s nejznámějšími a nejpoužívanějšími typy. V další sekci se pak věnujeme metodám, které pomocí posloupností uniformně rozložených náhodných čísel vytvářejí vzorky s jiným rozdělením pravděpodobnosti. Informace pro tuto kapitolu jsem čerpal zejména z [1, 5, 10, 11]

3.1 Lineární kongruentní generátor

Jedny z nejjednodušších a nejstarších generátorů pseudonáhodných čísel jsou lineární kongruentní generátory (LCG). Tyto generátory vytvářejí sekvence „náhodných celých čísel s uniformním rozdělením“ pomocí rekurentního vztahu:

$$x_n = (ax_{n-1} + c) \bmod M, \tag{3}$$

[1]Periodou rozumíme takové číslo $T \in \mathbb{N}$, že v posloupnosti pseudonáhodných vzorků platí pro každé uvažované $n \in \mathbb{N}$, že $x_{n+T} = x_n$.

kde $a \in \mathbb{N}$ se nazývá multiplikátorem, $c \in \mathbb{N} \cup \{0\}$ přírustkem. Parametr$M \in \mathbb{N}$ je modul, pomocí kterého se vytvářejí příslušné zbytkové třídy. Dalším důležitým parametrem generátoru je počáteční hodnota x_0, která se nazývá random seed („náhodné semínko“). Vhodná volba všech těchto parametrů je zásadní pro funkci generátoru, jelikož podstatně ovlivňují vlastnosti generovaných čísel a délku periody. Horní odhad velikosti (nejmenší) periody je M. Abychom dosáhli co nejlepších vlastností generovaných čísel a maximální (nejmenší) periody, je třeba se řídit následujícími pravidly:

1. c a M jsou navzájem nesoudělná čísla.
2. $a - 1$ je hodnota dělitelná všemi prvočíselnými faktory M.
3. $a - 1$ je číslo dělitelné čtyřmi, jestliže M je dělitelné čtyřmi.

Standardní generátory náhodných čísel generují čísla z intervalu $\langle 0, 1)$, toho docílíme vydělením všech hodnot x_i konstantou M.

Kvůli rychlé a výhodné implementaci je parametr M často volen jako mocnina čísla 2. Ukazuje se však, že při takto zvoleném parametru M nastává problém, kdy „nejméně významné bity korelují“, což může způsobit komplikace u některých typů simulací. Abychom těmto komplikacím předešli, je třeba jinak zvolit parametr M. Vhodnou náhradou je velké prvočíslo.

Příklad 3.1
Mějme LCG se zadanými parametry $a = 4, c = 15, m = 17$ a počáteční hodnotu $x_0 = 8$. Vygenerujme následujících 5 hodnot. Podle vztahu 3 získáme $x_1 = (4 \cdot 8 + 15) \bmod 17 = 13$, $x_2 = (4 \cdot 13 + 15) \bmod 17 = 16$, $x_3 = (4 \cdot 16 + 15) \bmod 17 = 11$, $x_4 = (4 \cdot 11 + 15) \bmod 17 = 8$, $x_5 = (4 \cdot 8 + 15) \bmod 17 = 13$. Všimněme si, že hodnota x_4 je stejná jako hodnota x_0. Kvůli nevhodně zvoleným parametrům má generátor malou periodu. ■

LCG není vhodný ke generování klíčů v kryptografii, jelikož je možné získat v polynomiálním čase vstupní parametry z několika pozorovaných výstupů. Ovšem i přes své nedostatky je LCG při správné volbě parametrů vhodný pro použití v klasických simulačních technikách.

3.2 Inverzní kongruentní generátor

Inverzní kongruentní generátor (ICG) je speciálním typem nelineárního generátoru. Pro generování čísel využívá tzv. modulární převrácenou hodnotu.

Definice 3.1 *Nechť je dáno číslo $c \in \mathbb{Z}$ a číslo $m \in \mathbb{N}$. Číslo $x \in \mathbb{Z}$ se nazývá modulární převrácenou hodnotou čísla c, pokud platí že $cx \equiv 1 \pmod{m}$.*

Příklad 3.2
Zvolme číslo $c = 7$ a $m = 3$, spočtemě modulární převrácenou hodnotu x čísla c. Víme, že musí platit vztah $cx \equiv 1 \pmod{m}$. Jednoduchou dedukcí dojdeme ke výsledku $x = 4$, jelikož $(4 \cdot 7) \bmod 3 = 1$. ■

Standardní vztah pro inverzní kongruentní generátor pak zapíšeme takto:

$$x_n = a\overline{x_{n-1}} + b \pmod{m}, \tag{4}$$

kde $\overline{x_{n-1}}$ značí výpočet modulární převrácené hodnoty x_{n-1}, $a \in \mathbb{N}$ je multiplikátor a $b \in \mathbb{N}$ je přírustek. Inverzní kongruentní generátor nemá narozdíl od lineárního kongruentního generátoru problémy s autokorelací, ale díky náročnosti výpočtu modulární převrácené hodnoty se ICG nestal příliš populárním.

3.3 Zpožděný Fibonacciho generátor

Zpožděný Fibonacciho generátor (LFG) se stal díky své jednoduchosti a rychlosti velice oblíbeným. Spadá do kategorie generátorů, které se snaží překonat klasický LCG. Generátor je pojmenován po Fibonacciho posloupnosti:

$$S_n = S_{n-1} + S_{n-2}, \quad S_0 = 1, S_1 = 1. \tag{5}$$

Uvedený vzorec byl zobecněn a tak vznikla skupina PRNG s následujícím přepisem:

$$X_i = (X_{i-p} \odot X_{i-q}) \bmod M, \tag{6}$$

kde $p \in \mathbb{N}$ a $q \in \mathbb{N}$ jsou tzv. skoky, $p > q$ a znak $\odot$ reprezentuje libovolnou binární aritmetickou operaci, jako například sčítání, odčítání nebo násobení. Stejně jako u LCG, je i u tohoto typu generátoru kvalita generovaných čísel závislá na správné volbě parametrů. Maximální velikost periody je určena volbou binární aritmetické operace, skoku p a číslem M. Empirické testy ukázaly, že vlastnosti generovaných čísel jsou nejlepší při použití binárního násobení[2], naopak operace XOR[3] dopadla nejhůře. LFG využívající

[2]Zde máme na mysli binární zápisy generovaných vzorků.

[3]XOR je logická (bitová) operace, jejíž hodnota je pravda, právě když každá vstupní hodnota nabývá, v porovnání s ostatními vstupy, unikátní hodnotu.

sčítání nebo odečítání patří mezi nejvíce používané, protože realizace uvedených operací je velmi rychlá a jednoduchá. Veškeré výpočty se provádí v pohyblivé řadové čárce, čímž se algoritmus vyhne náročnému převodu z celých čísel. Nesprávná volba skoků p a q je častou příčinou zhoršené kvality generovaných čísel. Například, pokud je číslo p velmi malé, tak generátor neprojde ani základními statistickým testy.

3.4 Mersenne Twister

Mersenne Twister (MT) je relativně nový generátor, jehož předností je velká perioda. Název je odvozen od tzv. Mersennova prvočísla, které udává velikost periody a lze zapsat ve tvaru $2^n - 1$, kde $n \in \mathbb{N}$. Standardní velikost periody je rovna $2^{19937} - 1$, tuto velikost využívají generátory testované v této práci. MT generátor byl navržen s hlavním cílem vyhnout se základním nedostatkům jiných generátorů, tedy malé periodě a rychlosti generování.

MT vytváří sekvenci bitových slov o velikosti w[4], které se reprezentují jako celá čísla z intervalu $\langle 0, 2^w - 1\rangle$ s uniformním rozdělením. Generátor je založen na následujícím rekurentním vztahu:

$$x_{k+n} = x_{k+m} \oplus [(x_k^u \mid\mid x_{k+1}^l)]A, \quad (k = 0, 1, ...), \tag{7}$$

kde zadané $n \in \mathbb{N}$ udává tzv. stupeň rekurence, m je zadané celé číslo z intervalu $\langle 1, n\rangle$, x_{k+m} je řádkový vektor, ve kterém je uloženo slovo o velikosti w, znak $\oplus$ je bitový operátor XOR a znak $||$ je bitový operátor OR. Aplikací horní a spodní bitové masky[5] o velikosti $(w - r)$ a r na x dostaneme x^u a x^l. Počáteční hodnoty jsou dány vektory $x_0, x_1, ..., x_{n-1}$. A je čtvercová matice velikosti $w \times w$ ve tvaru:

$$A = \begin{pmatrix} 0 & I_{w-1} \\ \alpha_{w-1} & (\alpha_{w-2}, ..., \alpha_0) \end{pmatrix} \tag{8}$$

kde I_{w-1} je jednotková matice velikosti $(w - 1) \times (w - 1)$ a vekor $\alpha = (\alpha_{w-1}, ..., \alpha_0)$ je zadaný a tvoří spodní vektor matice A. Násobení matice A vektorem x zleva je prováděno bitovým posunem tak, že:

$$xA = \begin{cases} x \gg 1, & \text{pokud } x_0 = 0, \\ (x \gg 1) \oplus \alpha, & \text{pokud } x_0 = 1, \end{cases} \tag{9}$$

[4]Bitové slovo je paměťová jednotka, která obsahuje w bitů.

[5]Bitová maska je vzor, podle kterého se změní zadané slovo. V našem případě aplikací bitové masky dostaneme slovo, které má část svých bitů nezměněnou a druhou část vynulovanou. Tento proces probíhá pomocí bitové operace AND.

$$\begin{array}{rr} & 10010110 \\ \text{XOR} & 00111011 \\ = & 10101101 \end{array}$$

(a) Bitová perace XOR

$$\begin{array}{r} 11010011 \\ = 01101001 \end{array}$$

(b) Bitový posun do prava

Obrázek 2: Ukázka bitových operací

kde x je vektor $(x_{w-1}, x_{w-2}, ..., x_0)$, znak $\oplus$ je bitový operátor XOR a znak $\gg$ je bitový posun doprava. Pokud by násobení matice A vektorem x zleva bylo prováděno klasickým způsobem, je třeba na jednotlivé prvky výsledného vektoru aplikovat $\mod 2$. MT je tedy parametrizován pomocí n, m, r, w a α a příslušnými počátečními hodnotami. Tento zápis vypadá složitě, ale je lehce pochopitelný z následujícího příkladu.

Příklad 3.3
Mějme jednoduchý MT generátor se zadanými parametry: $n = 3$, $m = 2$, $r = 3$, $w = 4$, $\alpha = (1, 1, 1, 1)$ a počáteční hodnoty $x_0 = 1011_2$, $x_1 = 1100_2$ a $x_2 = 0111_2$.Vypočtěme následující hodnotu x_3. Nejprveme aplikujeme bitové masky na vstupní hodnoty x_0 a x_1. Dostaneme $x_0^u = 1011_2 \,\&\, 1110_2 = 1010_2$ a $x_1^l = 1100_2 \,\&\, 0001_2 = 0000_2$ (znak $\&$ je bitový operátor AND). Dále vypočteme $x_0^u \,||\, x_1^l = 1010_2 \,||\, 0000_2 = 1010_2$. Násobení matice A a vektoru $x = (1, 0, 1, 0)$ je podle 9 určeno jako $x \gg 1 = (0, 1, 0, 1)$. Hledaná hodnota x_3 se pak vypočte jako $x_3 = 0111 \oplus 0101 = 0010_2$. ■

$$\begin{array}{rr} & 01101101 \\ \text{AND} & 00000111 \\ = & 00000101 \end{array}$$

(a) Dolni bitová maska

$$\begin{array}{rr} & 01101111 \\ \text{AND} & 11111000 \\ = & 01101000 \end{array}$$

(b) Horni bitová maska

Obrázek 3: Ukázka aplikace bitových mask

MT je první generátor umožňující rychlé generování vysoce kvalitních pseudonáhodných čísel a proto patří mezi nejrozšířenější PRNG. Jako defaultní PRNG je použit v programovacích jazycích PHP, Python, R a řadě dalších. Nevýhoda MT spočívá v jeho předvídatelnosti, takže není vhodný ke kryptografickým účelům. V ostatních aplikacích ovšem vykazuje dobré výsledky.

3.5 Blum Blum Shub

Blum Blum Shub (BBS) je generátor s dobrými kryptografickými vlastnostmi. Na rozdíl od předešlých generátorů vytváří sekvenci pseudonáhodných bitů. Je vytvořen na základně rekurentního vztahu:

$$x_n = x_{n-1}^2 \bmod M, \tag{10}$$

kde zadané M je součinem dvou velkých prvočísel p a q. Výstup je pak buď jeden či více méně významných bitů hodnoty x_n nebo paritní bit[6] x_n. Počáteční hodnota x_0 by měla být nesoudělná s číslem M. Ukazuje se, že prvočísla p a q je vhodné zvolit tak, aby byla kongruentní s $3 \pmod 4$.

Příklad 3.4
Mějme zadané $M = 115665149$ a počáteční hodnotu $x_0 = 1789456$. Vygenerujme sekvenci 10 bitů. Generované bity získáme jako 5 nejméně významných bitů vygenerovaných čísel. Nejdříve spočteme hodnoty $x_1 = 1789456^2 \bmod 115665149 = 78791020$ a $x_2 = 78791020^2 \bmod 115665149 = 77434588$. Vygenerované hodnoty převedeme do dvojkové soustavy, tedy $78791020_{10} = 1001011001001000000101101100_2$ a $77434588_{10} = 1001001110110001110110111100_2$. Pětice nejméně významných bitů jsou 01100 a 11100, vygenerovaná sekvence je tedy 0110011100. ■

BBS generátor není vhodný pro použití v simulacích, jelikož je extrémně pomalý. Jeho nejsilnější stránkou je obtížné předpovídání výstupních hodnot (které je způsobeno tzv. nepoddajností nalezení kvadratického rezidua).

Definice 3.2 *Nechť je dáno číslo $n \in \mathbb{N}$. Pak řekněme, že číslo $a \in \mathbb{Z}_n$ je kvadratickým reziduem s modulem n, pokud existuje číslo $b \in \mathbb{Z}_n$, takové, že $a \equiv b^2 \pmod n$.*

Bylo dokázano, že nalezení kvadratického rezidua je stejně náročné, jako rozluštění veřejného klíče kryptografického systému, který zahrnuje faktorizaci čísla s velkým počtem dělitelů. Při použití dostatečně velkého čísla M, pak nebude výstup obsahovat žádné „nenáhodné“ struktury odhalitelné v rozumném rozsahu počtu výpočtů.

[6] Paritní bit nabývá hodnoty 1 pokud je celkový počet jedniček bitové posloupnosti lichý a hodnoty 0 pokud ne.

3.6 Tausworthe generátor

Tausworthe generator (TG) je druh multiplikativního bitového generátoru. Je dán rekurentním vztahem:

$$x_n = (A_1 x_{n-1} + A_2 x_{n-2} + ... + A_k x_{n-k}) \bmod 2, \tag{11}$$

kde $x_i \in \{0, 1\}$, $A_i \in \{0, 1\}$ pro všechna $i \in 1, .., k$. Podobně jako předchozí generátor, TG také generuje sekvenci bitů, takže se používá převážně v kryptografických aplikacích. Pro jiné použití není díky své pomalosti vhodný. Pokud bychom potřebovali výstup v desítkové soustavě, můžeme každých k po sobě jdoucích bitů převést na decimální číslo.

Příklad 3.5
Mějme zadané $A_1 = 1$, $A_2 = 1$, $A_3 = 0$, $A_4 = 1$, $A_5 = 1$ a počáteční hodnoty $x_0 = 1$, $x_1 = 0$, $x_2 = 0$, $x_3 = 0$ a $x_4 = 1$. Vygenerujme následující bit. Hodnotu x_5 spočteme jako $x_5 = (A_1 x_4 + A_2 x_3 + A_3 x_2 + A_4 x_1 + A_5 x_0) \bmod 2 = (1 \cdot 1 + 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 1 \cdot 1) \bmod 2 = 1$.

■

4 Generátory pseudonáhodných čísel s obecným rozdělením pravděpodobnosti

V předchozí kapitole byly popsány generátory pseudonáhodných čísel, jejichž výstupem je posloupnost čísel s uniformním rozdělením. V praxi je často nutné vygenerovat vzorek s jiným, než uniformní rozdělením, proto vznikly speciální postupy k tomu určené. Základní metody jsou probrány v této kapitole. Při psaní kapitoly jsem čerpal z [2].

4.1 Metoda inverzní transformace

Metoda inverzní transformace je založena na následujícím jednoduchém tvrzení:

Věta 4.1 *Nechť je dána rostoucí distribuční funkce F libovolné náhodné veličiny. Označme symbolem F^{-1} její inverzi. Nechť U je náhodná veličina, která má uniformní rozdělení $\langle 0, 1\rangle$. Definujme náhodnou veličinu $X = F^{-1}(U)$. Pak X má rozdělení pravděpodobnosti s distribuční funkcí F.*

Důkaz. Pro každé $x \in \mathbb{R}$ platí:

$$P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x).$$

■

Na základě předešlé věty lze ke každému vygenerovanému vzorku u náhodné veličiny U přiřadit vzorek $F^{-1}(u)$. Tento vzorek pak bude realizací náhodné veličiny X, jejíž distribuční funkcí je F. Při praktické realizaci metody je třeba zvážit numerickou náročnost výpočtu hodnot inverzní funkce F^{-1}.

Příklad 4.1
Pokud chceme generovat vzorky náhodné veličiny s distribuční funkcí $F(x) = \frac{1}{2} + \frac{1}{\pi}\mathrm{arctg}x$, stačí generovat vzorky u náhodné veličiny U s uniformním rozdělením $(0, 1)$ a dopočítat požadované hodnoty $\mathrm{tg}(\pi u - \frac{\pi}{2})$. ■

Poznámka 4.1 V případě, že F není rostoucí funkcí, lze místo inverzní funkce k F použít její vhodné zobecnění dané předpisem

$$\widetilde{F}^{-1}(u) = \inf\{x \in \mathbb{R} : u \leq F(x)\}.$$

4.2 Metoda přijetí-zamítnutí vzorku

Mějme k dispozici generátor vzorků náhodné veličiny U s uniformním rozdělením na $\langle 0, 1\rangle$. Dále předpokládejme, že jsme schopni generovat vzorky náhodné veličiny Y s rozdělením pravděpodobnosti daným hustotou g a distribuční funkcí G

$$G(y) = \int_{-\infty}^{y} g(s)\mathrm{d}s.$$

Naším cílem je nalézt postup, který pro zadanou nezápornou funkci $h : \mathbb{R} \to \mathbb{R}$, takovou že $\int_{-\infty}^{\infty} h(x)\mathrm{d}x = 1$ vybere ze vzorků Y jen některé, a to tak, že provedený výběr bude odpovídat realizaci náhodné veličiny X s hustotou pravděpodobnosti h. Dříve než uvedeme příslušný algoritmus, formulujme důležité předpoklady. Požadujeme:

- $$\sup_{s\in\mathbb{R}} \frac{h(x)}{g(x)} \leq c \in \mathbb{R}^+$$
- U, Y jsou nezávislé veličiny.

Samotný algoritmus pak vypadá následovně:

1. Generujeme vzorek g náhodné veličiny Y s distribuční funkcí G.
2. Generujeme vzorek u náhodné veličiny U, nazávislé na Y.
3. Pokud
$$u \leq \frac{h(y)}{c \cdot g(y)},$$
přijmeme číslo y jako vzorek náhodné veličiny X. Jinak vzorek zahodíme. Dále pokračujeme k bodu 1.

Věta 4.2 *Postupem popsaným v algoritmu získáme vzorky náhodné veličiny X s rozdělením pravděpodobnosti daném hustotou h.*

Poznámka 4.2 Před provedením důkazu upozorněme, že protože Y je náhodná veličina, také $h(Y), g(Y)$, a proto i $\frac{h(Y)}{c\cdot g(Y)}$ jsou náhodné veličiny, přičemž platí odhad

$$0 \leq \frac{h(Y)}{c \cdot g(Y)} \leq 1.$$

Důkaz. Popsaný algoritmus zřejmě poskytuje vzorky náhodné veličiny X, která představuje náhodnou veličinu Y, za podmínky $U \leq \frac{h(Y)}{c \cdot g(Y)}$. Musíme dokázat, že

$$P(X \leq x) = H(x), \text{ kde } H(x) = \int_{-\infty}^{\infty} h(s)\mathrm{d}s.$$

Podle Bayesova vzorce lze postupně psát:

$$P(X \leq x) = P(Y \leq x | U \leq \frac{h(Y)}{c \cdot g(Y)}) = \frac{P(U \leq \frac{h(Y)}{c \cdot g(Y)} | Y \leq x) \cdot P(Y \leq x)}{P(U \leq \frac{h(Y)}{c \cdot g(Y)})}, \tag{12}$$

přitom zřejmě:

$$P(Y \leq x) = G(x),$$

$$P(U \leq \frac{h(Y)}{c \cdot g(Y)}) = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\frac{h(y)}{c \cdot g(y)}} f_{UY}(u, y)\mathrm{d}u \right) \mathrm{d}y,$$

kde f_{UV} je sdružená distribuční funkce náhodného vektoru (U, Y). Z nezávislosti U, Y přitom plyne, že na $\langle 0, 1\rangle \times \mathbb{R}$:

$$f_{UY}(u, y) = f_U(u) \cdot f_Y(y) = 1 \cdot g(y),$$

protože U má konstatní hustotu 1 a marginální hustotou f_Y sdružené hustoty f_{UV} je funkce g. Proto

$$P(U \leq \frac{h(Y)}{c \cdot g(Y)}) = \int_{-\infty}^{\infty} \left(g(y) \cdot \int_{0}^{\frac{h(y)}{c \cdot g(y)})} 1\mathrm{d}u \right) \mathrm{d}y = \int_{-\infty}^{\infty} g(y) \cdot \frac{h(y)}{c \cdot g(y)} \mathrm{d}y = \frac{1}{c} \int_{-\infty}^{\infty} h(y)\mathrm{d}y = \frac{1}{c}.$$

Konečně,

$$P(U \leq \frac{h(Y)}{c \cdot g(Y)} | Y \leq x) = \frac{P(U \leq \frac{h(Y)}{c \cdot g(Y)}, Y \leq x)}{P(Y \leq x)} = \frac{\int_{-\infty}^{x} \left(\int_{-\infty}^{\frac{h(y)}{c \cdot g(y)}} f_{UY}(u, y)\mathrm{d}u \right) \mathrm{d}y}{G(x)} =$$

$$= \frac{\int_{-\infty}^{x} \left(f_Y(y) \int_{-\infty}^{\frac{h(y)}{c \cdot g(y)}} f_U(u)\mathrm{d}u \right) \mathrm{d}y}{G(x)} = \frac{\int_{-\infty}^{x} \left(g(y) \int_{0}^{\frac{h(y)}{c \cdot g(y)}} 1\mathrm{d}u \right) \mathrm{d}y}{G(x)} = \frac{1}{c \cdot G(x)} \cdot \int_{-\infty}^{x} h(y)\mathrm{d}y =$$

$$= \frac{1}{c} \cdot \frac{H(x)}{G(x)}.$$

Odtud po dosazení do pravé strany v 12 plyne:

$$P(X \leq x) = \frac{\frac{1}{c} \cdot \frac{H(x)}{G(x)} \cdot G(x)}{\frac{1}{c}} = H(x).$$

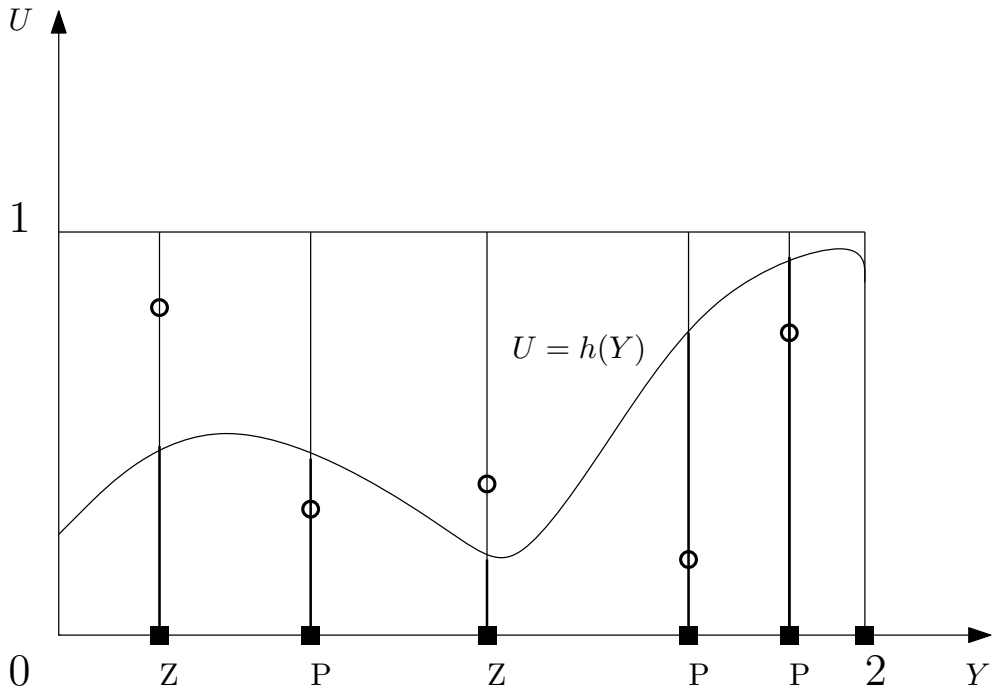
■

Poznámka 4.3 Při jedné realizaci algoritmu je pravděpodobnost přijetí vzorku dána hodnotou $P(U \leq \frac{h(Y)}{c \cdot g(Y)}) = \frac{1}{c}$, kterou jsme vypočetli v předešlém důkazu. Pravděpodobnost prvního přijetí vzorku v n-té realizaci je proto $(1 - \frac{1}{c})^{n-1} \cdot \frac{1}{c}$. Odtud lze odvodit, že střední hodnota do prvního přijetí vzorku je c. Snažíme se proto volit hustot g náhodné veličiny Y tak, aby byla co nejblíž f, tedy, aby c bylo co nejmenší. K tomu lze někdy použít metodu inverze distribuční funkce.

Poznámka 4.4 Podobně jako v předchozím případě, lze zformulovat a dokázat algoritmus pro případ diskrétní náhodné veličiny. Nechť p je požadovaná pravděpodobnostní funkce diskrétní veličiny.

1. Generujeme vzorek y diskrétní náhodné veličiny Y s pravděpodobnostní funkcí q.
2. Generujeme vzorek u náhodné veličiny U s uniformním rozdělením na $(0, 1)$.
3. Pokud
$$u \leq \frac{p(y)}{c \cdot q(y)}, \quad \text{kde } \sup_k \frac{p(k)}{q(k)} = c \in \mathbb{R}^+,$$
vzorek y přijmeme. V opačném případě jej zamítneme a vrátíme se ke kroku 1.

Poznámka 4.5 Postup algoritmu lze snadno modifikovat pro případ, že hustota g je nenulová pouze na konečném intervalu $(a, b) \subset \mathbb{R}$. Ve speciálním případě lze metodu přijetí-zamítnutí dobře graficky ilustrovat:



Obrázek 4: Ukázka metody zamítání vzorků

Obrázek zachycuje 5 vzorků náhodného vektoru (U, Y) vygenerovaných s rovnoměrným rozdělením na $[0, 1] \times [0, 2]$. Vzorky označené P jsou přijaty a tvoří výsledný výstup. Ostatní vzorky Y označené Z jsou zamítnuty.

5 Testovací baterie

Testování generátorů je poměrně obsáhlé téma, jelikož existuje celá řada statistických testů, které jsou vhodné pro ověření jejich kvality. Je důležité si uvědomit, že cílem těchto testů není je vyvrátit, že je generátor nekvalitní. Tento přístup je zapříčiněn samotnou definicí náhodnosti, protože determinisitcky zkonstruovat něco, co je náhodné, je prakticky nemožné. Proto se tyto testy zaměřují na hledání vad a vzorů ve vygenerované sekvenci a na základně jejich výskytu generátor ohodnotí.

Kvůli rozšíření a důležitosti PRNG jsou kladeny vysoké nároky na jejich kvalitu. Z toho důvodu vzniklo několik testovacích baterií. V této práci je použita testovací baterie NIST, která je veřejně dostupná a není předmětem žádných ochran autorských práv. Tato baterie testuje sekvence náhodných bitů o velikosti n. Další testovací baterie jsou DieHard a TestU01. Tyto uvedené baterie se od sebe odlišují, ale řada jejich testů je založená na stejném principu nebo jsou dokonce totožné.

V této kapitole budou popsány jednotlivé testy z testovací baterie NIST. Detailní technický popis všech testů lze nalézt v [3].

5.1 Matematický základ

Řada testů v této testovací baterii využívá vlastností standardního normálního a chí-kvadrát (χ^2) rozdělení. Pokud je testovaná sekvence nenáhodná, získaná testová statistika se vyskytne v extrémních oblastech odkazovaného rozdělení. Testová statistika pro standardní normální rozdělení je ve tvaru $z = \frac{x-\mu}{\sigma}$, kde x je hodnota testové statistiky vzorku, μ a σ^2 jsou třední hodnoy a rozptyl. χ^2 rozdělení je použito ke srovnání pomocí testu dobré shody. Testová statistika je ve tvaru $\chi^2 = \sum \frac{(o_i - e_i)^2}{e_i}$, kde o_i je zjištěná absolutní četnost a e_i je očekávaná četnost.

Speciální funkce použité v testech jsou pak tzv. doplňková chybová funkce

$$\mathrm{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty e^{-u^2} \mathrm{du}, \tag{13}$$

distribuční funkce normovaného normálního rozdělení

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-u^2/2} \mathrm{du}, \tag{14}$$

a gama funkce:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} \mathrm{dt}. \tag{15}$$

Z gama funkce vychází definice tzv. spodní neúplné gama funkce

$$\mathrm{P}(a,x) = \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} \mathrm{dt}, \tag{16}$$

a horní neúplné gama funkce

$$\mathrm{Q}(a,x) = 1 - P(a,x) = \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1} \mathrm{dt}. \tag{17}$$

Uvedené funkce se využívají k vyjádření p-hodnot u jednotlivých testů.

5.2 Frekvenční test

Frekvenční test se zaměřuje na podíl nul a jedniček v testované sekvenci. Cílem testu je zjistit, zda je počet nul a jedniček přibližně stejný, podobně jako u skutečně náhodné sekvence.

Test využívá aproximaci binomického rozdělení pomocí normálního rozdělení.

Věta 5.1 (Lévyho-Lindebergova věta) *Jestliže $X_1, X_2, ..., X_n$ jsou nezávislé náhodné veličiny se stejným rozdělením, stejnou střední hodnotou μ a stejným konečným rozptylem σ^2, pak pro normovanou náhodnou veličinu*

$$U_n = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sqrt{n\sigma^2}} \tag{18}$$

platí vztah

$$\lim_{n\to\infty} P(U_n < u) = \Phi(u), \tag{19}$$

kde $\Phi(u)$ je distribuční funkce normovaného normálního rozdělení $\mathrm{N}(0,1)$.

Odkazované rozdělení pro testovou statistiku je polo-normální.

Poznámka 5.1 (Polo-normální rozdělení) Nechť X je normální rozdělení $\mathrm{N}(0,\sigma^2)$, pak $Y = |X|$ je polo-normální rozdělení.

Test nezajímá na kterou „stranu“ se vychýlí testované hodnoty, ale v jaké míře. Proto je po převodu hodnot na -1 a 1 použito polo-normální rozdělení. V prvním kroku testu se vstupní sekvence převede na hodnoty -1 a 1. Tím získáme realizace nezávislé náhodné veličiny se střední hodnotou $\mu = (-1 \cdot 0,5) + (1 \cdot 0,5) = 0$ a rozptylem $\sigma^2 = \frac{1}{2}(1-0)^2 +$

$\frac{1}{2}(-1-0)^2 = 1$. Aplikací Lévyho-Lindebergovy věty pak dostaneme testovou statistiku danou předpisem:

$$S_{obs} = \frac{|S_n|}{\sqrt{n}}, \tag{20}$$

kde $|S_n|$ je absolutní hodnota součtu všech prvků testované sekvence a n je počet prvků. p-hodnota se získá jako hodnota funkce

$$\mathrm{erfc}\left(\frac{S_{obs}}{\sqrt{2}}\right). \tag{21}$$

Poznámka 5.2 (Odvození výpočtu p-hodnoty) Nechť X je náhodná veličina s normálním rozdělením $N(0,1)$ a distribuční funkcí

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-u^2/2}\mathrm{du}.$$

Mějme veličinu $\widetilde{X} = \frac{X}{\sqrt{2}}$ a její distribuční funkci

$$\Phi_{\widetilde{X}}(\tilde{x}) = P(\widetilde{X} < \tilde{x}) = P(X < \tilde{x} \cdot \sqrt{2}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\tilde{x}\sqrt{2}} e^{-u^2/2}\mathrm{du}.$$

Pomocí substituce φ: $\omega = \frac{u}{\sqrt{2}}$, $\mathrm{d}\omega = \frac{\mathrm{du}}{\sqrt{2}}$, $\mathrm{u} = -\infty \to \omega = -\infty$, $u = \tilde{x}\sqrt{2} \to \omega = \tilde{x}$ upravme tuto distribuční funkci tak, že:

$$\begin{aligned}\Phi_{\widetilde{X}}(\tilde{x}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\tilde{x}\sqrt{2}} e^{-u^2/2}\mathrm{du} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\tilde{x}\sqrt{2}} e^{-\left(\frac{u}{\sqrt{2}}\right)^2}\mathrm{du} = \\ = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\tilde{x}} e^{-\omega^2}\sqrt{2}\mathrm{d}\omega = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\tilde{x}} e^{-\omega^2}\mathrm{d}\omega.\end{aligned}$$

Nyní mějme veličinu $Y = |\widetilde{X}|$ a její distribuční funkci

$$\Phi_Y(y) = \begin{cases} \frac{1}{\sqrt{\pi}} \int_{-y}^{y} e^{-\omega^2}\mathrm{d}\omega, & \text{pro } y > 0 \\ 0, & \text{pro } y \leq 0. \end{cases}$$

Pro výpočet p-hodnoty pak platí:

$$\text{p-hod.} = 1 - \frac{1}{\sqrt{\pi}} \int_{-y}^{y} e^{-\omega^2}\mathrm{d}\omega = 1 - \frac{1}{\sqrt{\pi}} \left(\int_{-\infty}^{\infty} e^{-\omega^2}\mathrm{d}\omega - 2\int_{y}^{\infty} e^{-\omega^2}\mathrm{d}\omega \right) = \frac{2}{\sqrt{\pi}} \int_{y}^{\infty} e^{-\omega^2}\mathrm{d}\omega,$$

což je předpis použité funkce $\mathrm{erfc}(y)$, kde $y = \frac{S_{obs}}{\sqrt{2}}$.

5.3 Blokový frekvenční test

Blokový frekvenční test se zaměřuje na podíl jedniček v blocích o velikosti M. Cílem testu je zjistit, zda absolutní četnosti jsou přibližně $M/2$, jak by se dalo očekávat u skutečně náhodné sekvence.

Testovaná sekvence se rozdělí do N bloků (buněk) o velikosti M. Je použit tzv. test dobré shody (Pearsonův chí-kvadrát test), který umožňuje ověřit, zda má náhodná veličina určité, předem dané rozdělení. Je dán předpisem:

$$\chi^2 = \sum_{i=1}^{N} \frac{(O_i - E_i)^2}{E_i}, \tag{22}$$

kde O_i jsou absolutní četnosti a E_i očekávané absolutní četnosti pro jednotlivé buňky $i = 1, .., N$. Důležitým parametrem testu dobré shody je počet tzv. stupňů volnosti, který je roven $n - 1$. Tato hodnota udává počet pozorovaných nezávislých veličin, na kterých je založen parametrický odhad a slouží k určení tabelovaných p-hodnot. Po úpravě předpisu testu dobré shody pouze pro rozdělení jedniček dostaneme testovou statistiku:

$$\chi^2(\text{obs}) = 4M \sum_{i=1}^{N} \left[\pi_i - \frac{1}{2}\right], \tag{23}$$

kde π_i jsou relativní četnosti jedniček v jednotlivých blocích, M je velikost bloků, N je celkový počet bloků a $\frac{1}{2}$ udává očekávanou relativní čenost jedniček. p-hodnota se získá jako hodnota funkce

$$\frac{\int_{\chi^2(obs)}^{\infty} e^{-u/2} u^{N/2-1} du}{\Gamma(N/2)2^{N/2}} = \frac{\int_{\chi^2(obs)/2}^{\infty} e^{-u} u^{N/2-1} du}{\Gamma(N/2)} = \text{igamc}\left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right) \tag{24}$$

5.4 Test sérií

Test sérií se zaměřuje na celkový počet sérií v testované sekvenci. Za sérii přitom považujeme nepřerušený řetězec stejných bitů. Série délky k je řetězec k stejných bitů ohraničených bity s jinou hodnotou. Cílem testu je zjistit, zda počet sérií nul a jedniček různých délek je podobný jako u skutečně náhodné sekvence.

Test je založen na rozdělení celkového počtu sérií V_n. Pro pevně daný poměr $\pi_n = \sum_{i=1}^{n} \varepsilon_i / n$ (kde ε_i jsou jednotlivé prvky z testované posloupnosti), který by měl být blízko $1/2$, platí:

$$\lim_{n\to\infty} P\left(\frac{V_n - 2n\pi(1-\pi)}{2\sqrt{n}\pi(1-\pi)} \leq z\right) = \Phi(z), \tag{25}$$

kde $\Phi(z)$ je distribuční funkce normovaného normálního rozdělení. p-hodnota se získá jako hodnota funkce:

$$\mathrm{erfc}\left(\frac{|V_n(obs) - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right). \tag{26}$$

5.5 Blokový test nejdelší série

Blokový test nejdelší série se zaměřuje na nejdelší sérii jedniček v blocích o velikosti M. Cílem testu je zjistit, zda délka nejdelší série jedniček je konzistentní s délkou série jedniček skutečně náhodné sekvence.

Testovaná sekvence se disjunktně rozdělí do N bloků velikosti M.V každém bloku se nalezne nejdelší série jedniček délky l_j. Četnosti v_i všech těchto délek se rozdělí do $K+1$ tříd. Rozdělení do těchto tříd probíhá na základě délky l_j, která je reprezentována danou četností. Každá třída pak přiřadí četnostem jejich teoretickou pravděpodobnost výskytu π_i. Počet tříd je určen velikostí bloků M. Počet tříd a teoretické pravděpodobnost jsou zapsány v příloze A.1. Testová statistika je dána předpisem:

$$\chi^2(\mathrm{obs}) = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)}{N\pi_i}, \tag{27}$$

kde v_i jsou zjištěné četnosti v daných třídách a π_i jsou jejich teoretické pravděpodobnosti výskytu. p-hodnota se získá jako hodnota funkce

$$\frac{\int_{\chi^2(obs)}^{\infty} e^{-u/2}u^{K/2-1}du}{\Gamma(K/2)2^{K/2}} = \mathrm{igamc}\left(\frac{K}{2}, \frac{\chi^2(\mathrm{obs})}{2}\right). \tag{28}$$

5.6 Test hodnosti binární matice

Test hodnosti binární matice se zaměřuje na hodnosti jednotlivých podmatic z testované sekvence. Cílem testu je zjistit, zda existuje lineární závislost mezi podřetězci originální sekvence.

Testovaná sekvence se rozdělí do N disjuknтních matic o rozměrech $M \times Q$ a určí se jejich hodnost (R_l). Testová statistika je dána předpisem:

$$\chi^2(\mathrm{obs}) = \frac{(F_M - 0.2888N)^2}{0.2888N} + \frac{(F_{M-1} - 0.5776N)^2}{0.5776N} + \frac{(N - F_M - F_{M-1} - 0.1336N)^2}{0.1336N}, \tag{29}$$

kde F_M je počet matic s plnou hodností[7], F_{M-1} je počet matic s hodností o jednu menší a $N - F_M - F_{M-1}$ je počet zbylých matic. p-hodnota se získá jako hodnota funkce

$$\mathrm{e}^{-\chi^2(\mathrm{obs})/2}. \tag{30}$$

5.7 Test shody nepřekrývajících se řetězců

Test shody nepřekrývajících se řetězců se zaměřuje na výskyt předem specifikovaných řetězců v testované sekvenci. Cílem testu je zjistit, zda generátor vytváří příliš velké množství těchto neperiodických vzorů. Tento test využívá m-bitové okno pro nalezení specifických m-bitových vzorů. Pokud není hledaný vzor nalezen, okno se posune o jeden bit dál v prohledávané sekvenci. Při nalezení vzoru se okno posune na bit za nalezeným řetězcem.

Testovaná sekvence se rozdělí do N bloků velikosti M. Pro daný vzor B délky m se zjistí četnost výskytu W_j hledaného vzoru B v bloku j. Za předpokladu náhodnosti platí, že střední hodnota $\mu = (M - m + 1)/2^m$ a rozptyl $\sigma^2 = M\left(\frac{1}{2^m} - \frac{2m-1}{2^{2m}}\right)$. Testová statistika je daná předpisem:

$$\chi^2(\mathrm{obs}) = \sum_{j=1}^{N} \frac{(W_j - \mu)^2}{\sigma^2}, \tag{31}$$

kde W_j jsou četnosti výskytu vzoru B v daných blocích, μ je střední hodnota a σ^2 je odchylka. p-hodnota se získá jako hodnota funkce

$$\mathrm{igamc}\left(\frac{N}{2}, \frac{\chi^2(\mathrm{obs})}{2}\right). \tag{32}$$

5.8 Test shody překrývajících se řetězců

Test shody překrývajících se řetězců se zaměřuje na výskyt předem specifikovaných řetězců v testované sekvenci. Podobně jako předchozí test, i tento využívá m-bitové okno pro vyhledávání specifických m-bitových vzorů. Pokud není hledaný vzor nalezen, okno se posune o jeden bit dál v prohledávané sekvenci. Při nalezení vzoru se, na rozdíl od předchozího testu okno posune na následující bit, čímž je zajištěno nalezení i překrývajících se řetězců.

Testovaná sekvence se rozdělí do N bloků velikosti M. Vypočte se četnost výskytů vzoru B délky m v jednotlivých blocích. Počet výskytů hledaného vzoru se zaznamená

[7] Pokud matice A s rozměry $M \times Q$, kde $M \geq Q$ má hodnost rovnou Q, pak o matici A říkáme, že má plnou hodnost.

inkremetancí správné buňky v poli v_i (kde $i = 0, .., 5$), takovým způsobem, že při žádném výskytu vzoru B v jednom bloku se navyší v_0, při jednom výskytu vzoru B se navýší v_1 atp. Při pěti či více výskytech vzoru B se navýší buňka v_5. Testová statistika je dána předpisem:

$$\chi^2(obs) = \sum_{i=0}^{5} \frac{(v_i - N\pi_i)^2}{N\pi_i}, \tag{33}$$

kde v_i jsou počty bloků s četností výskytu i vzoru B a π_i jsou jejich teoretické pravděpodobnosti. p-hodnota se získá jako hodnota funkce

$$\mathrm{igamc}\left(\frac{5}{2}, \frac{\chi^2(\mathrm{obs})}{2}\right). \tag{34}$$

5.9 Maurerův univerzální statistický test

Maurerův univerzální statistický test se zaměřuje na počet bitů mezi stejnými vzory v testované sekvenci. Cílem testu je v podstatě zjistit, zda testovaná sekvence může být efektivně zkomprimována bez ztráty dat. Vysoce komprimovatelná sekvence se přirozeně považuje za nenáhodnou.

Testovaná sekvence délky n se rozdělí do dvou segmentů. První segment se skládá z Q L-bitových bloků a slouží k inicializaci testu. Druhý segment se skládá z K L-bitových bloků a slouží k samotnému testování. Test postupně bere L-bitové bloky z druhého segmentu a napříč celou sekvencí hledá nejbližší stejný předchozí L-bitový blok. Index nejbližšího stejného bloku se zapíše do prvku T_j pole T. Testová statistika je dána předpisem:

$$f_n = \frac{1}{K} \sum_{i=Q+1}^{Q+K} \log_2(i - T_j), \tag{35}$$

kde $i - T_j$ jsou vzdálenosti stejných bloků, Q je počet bloků v prvním segmentu a K je počet bloku v segmentu druhém. P-hodnota se získá ze vztahu:

$$\mathrm{erfc}\left(\left|\frac{f_n - \mathrm{expectedValue}(L)}{\sqrt{2}\sigma}\right|\right) \tag{36}$$

kde $\mathrm{expectedValue}(L)$ a odchylka σ jsou hodnoty z tabulky předpočítaných hodnot:

L	expectedValue	odchylka σ	L	expectedValue	odchylka σ
6	5,2177052	2,954	12	11,168765	3,401
7	6,1962507	3,125	13	12,168070	3,410
8	7,1836656	3,238	14	13,167693	3,416
9	8,1764248	3,311	15	14,167488	3,419
10	9,1723243	3,356	16	15,167379	3,421
11	10,170032	3,384			

5.10 Sériový test

Sériový test se zaměřuje na četnosti výskytu všech možných m-bitových překrývajících se řetězců. Cílem testu je zjistit, zda četnost výskytů m-bitových řetězců je přibližně stejná, jako u skutečně náhodné sekvence. Náhodné sekvence přitom maji uniformní rozdělení, tedy každý m-bitový řetězec má stejnou šanci výskytu.

V prvním kroku se z testované sekvence vytvoří rozšířená sekvence ε'. Rozšíření se provede přidáním prvních $m-1$ bitů na konec originální sekvence. Určí se četnosti výskytu všech překrývajících se m-bitových, $(m-1)$-bitových a $(m-2)$-bitových řetězců. Nechť $v_{i_1 \ldots i_m}$ zaznamenají četnosti všech m-bitových řetězců $i_1 \ldots i_m$. Nechť $u_{i_1 \ldots i_{m-1}}$ zaznamenají frekvence všech $(m-1)$-bitových řetězců $i_1 \ldots i_{m-1}$. Nechť $t_{i_1 \ldots i_{m-2}}$ zaznamenají frekvence všech $(m-2)$-bitových řetězců $i_1 \ldots i_{m-2}$. Dále se určí:

$$\begin{aligned} \psi_m^2 &= \frac{2^m}{n} \sum_{i_1 \ldots i_m} v_{i_1 \ldots i_m}^2 - n, \\ \psi_{m-1}^2 &= \frac{2^{m-1}}{n} \sum_{i_1 \ldots i_{m-1}} u_{i_1 \ldots i_{m-1}}^2 - n, \\ \psi_{m-2}^2 &= \frac{2^{m-2}}{n} \sum_{i_1 \ldots i_{m-2}} t_{i_1 \ldots i_{m-2}}^2 - n, \end{aligned} \tag{37}$$

kde n je počet bitů originální sekvence. Testové statistiky jsou pak dány vztahy:

$$\begin{aligned} \nabla \psi_m^2(obs) &= \psi_m^2 - \psi_{m-1}^2, \\ \nabla^2 \psi_m^2(obs) &= \psi_m^2 - 2\psi_{m-1}^2 + \psi_{m-2}^2, \end{aligned} \tag{38}$$

kde $\nabla\psi_m^2(obs)$ a $\nabla^2\psi_m^2(obs)$ vyjadřují, jak dobře pozorované četnosti odpovídají očekávaným četnostem m-bitových řetězců. p-hodnoty jsou získány jako hodnoty funkcí

$$\begin{aligned} &\mathrm{igamc}\left(2^{m-2}, \nabla\psi_m^2(\mathrm{obs})\right), \\ &\mathrm{igamc}\left(2^{m-3}, \nabla^2\psi_m^2(\mathrm{obs})\right). \end{aligned} \tag{39}$$

5.11 Test přibližné entropie

Test přibližné entropie se jako předchozí test zaměřuje na četnost výskytu všech možných m-bitových překrývajících se řetězců. Cílem testu je porovnání četností dvou po sobě jdoucích řetězců délky m a $m+1$ s jejich očekávanými četnostmi.

Jako v předchozím testu se testovaná sekvence rozšíří o prvních $m-1$ bitů. Určí se četnost výskytu všech m-bitových řetězců a vypočítají se jejich teoretické pravděpodobnosti výskytu π_i (kde $i = 0..2^m - 1$). Dále se vypočítá hodnota $\varphi^{(m)} = \sum_{i=0}^{2^m-1} \pi_i \log \pi_i$. Celý tento proces se opakuje také pro $(m+1)$-bitové řetězce. Testová statistika je pak dána předpisem:

$$\chi^2(\mathrm{obs}) = 2n[\log 2 - (\varphi^{(m)} - \varphi^{(m+1)})], \tag{40}$$

kde $\varphi^{(m)} - \varphi^{(m+1)}$ vyjadřuje, jak pozorované frekvence souhlasí s očekávanými frekvencemi m-bitových a $(m+1)$-bitových řetězců. p-hodnota se získá jako hodnota funkce

$$\mathrm{igamc}\left(2^{m-1}, \frac{\chi^2(\mathrm{obs})}{2}\right). \tag{41}$$

5.12 Test kumulativních součtů

Test kumulativních součtů se zaměřuje na maximální odchylku součtů hodnot testované sekvence. Cílem testu je zjistit, zda kumulativní součty částečných sekvencí jsou příliš malé nebo velké v porovnání s kumulativními součty skutečně náhodné sekvence.

Testovaná sekvence se převede na hodnoty -1 a 1. Vypočítají se částečné součty S_i všech podsekvencí originální sekvence. Test nabízí dva módy 0 a 1. Pro mód 0 se částečné součty získají ze vztahu $S_k = S_{k-1} + X_k$, kde $S_1 = X_1$, $S_2 = X_1 + X_2$ atd. Pro mód 1 se částečné součty získají ze vztahu $S_k = S_{k+1} + X_{n-k+1}$, kde $S_1 = X_n$, $S_2 = X_n + X_{n-1}$ atd. Testová statisika je dána předpisem:

$$z\ (\mathrm{obs}) = \max_{1 \le k\ \le n} |S_k| \tag{42}$$

kde $\max_{1\leq k\ \leq n}|S_k|$ je největší z absolutních hodnot částečných součtů. p-hodnota se získá jako hodnota funkce

$$1 - \sum_{k=(\frac{-n}{z}+1)/4}^{(\frac{n}{z}-1)/4} \left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}}\right)\right] + \\ + \sum_{k=(\frac{-n}{z}-3)/4}^{(\frac{n}{z}-1)/4} \left[\Phi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right)\right]. \tag{43}$$

Popišme ještě stručně další testy obsažené v baterii NIST.

5.13 Spektrální test (rychlá Fourierova transformace)

Spektrální test se zaměřuje na nejvyšší vrcholy diskrétní Fourierovy transformace vstupní sekvence. Cílem testu je objevit opakující se vzory v testované sekvenci, které by indikovaly odchylky od předpokládané náhodnosti. Záměrem je zjistit, zda počet vrcholů přesahujících 95% hranici je významně vyšší než $5\ \%$.

V prvním kroku se originální sekvence převede na hodnoty -1 a 1. Na takto převedenou sekvenci se aplikuje diskrétní Fourierova transformac. Aplikací diskrétní Fourierovy transformace vznikne sekvence S komplexních proměnných, která reprezentuje periodické složky sekvence bitů s různou frekvencí. Vypočte se $M = \text{modulus}(S')$, kde S' je podřetězec skládající se z prvních $n/2$ prvků S a funkce modulus vytváří sekvenci nejvyšších vrcholů. Dále se vypočte hranice $T = \sqrt{(\ln \frac{1}{0.05})n}$, kterou by za předpokladu náhodnosti nemělo překročit $95\ \%$ hodnot. Testová statistika je pak dána předpisem:

$$d = \frac{N_1 - N_0}{\sqrt{0.95 \cdot 0.05 \cdot n/4}}, \tag{44}$$

kde N_1 je počet vrcholů z M, které nepřesáhly hranici T a N_0 je očekávaný počet vrcholů $(0.95 \cdot n/2)$, které nepřesáhly hranici T. p-hodnota se získá jako hodnota funkce

$$\text{erfc}\left(\frac{|d|}{\sqrt{2}}\right). \tag{45}$$

5.14 Test linerání složitosti

Test lineární složitosti se zaměřuje na délku lineárního zpětnovazebného posuvného registru (LFSR). Cílem testu je rozhodnout, zda je testovaná sekvence dostatečně komplexní,

na to aby se dala považovat za náhodnou. Náhodné sekvence mají delší LSFR. Příliš krátké LSFR poukazují na nenáhodnost.

Testovaná sekvence se rozdělí na N disjunktních bloků velikosti M. Pomocí Berlekampova Masseyho algoritmu [7] se určí lineární složitost L_i všech bloků. Za předpokladu náhodnosti se vypočítá střední hodnota $\mu = \frac{M}{2} + \frac{9+(-1)^{M+1}}{36} - \frac{M/3+2/9}{2^M}$. Pro jednotlivé bloky se určí hodnota $T_i = (-1)^M \cdot (L_i - \mu) + 2/9$. Vypočtené hodnoty T_i inkrementují třídy v_j (kde $j = 0, .., 6$) podle pravidel zapsaných v příloze A.2. Testová statistika je dána předpisem:

$$\chi^2(\text{obs}) = \sum_{i=0}^{6} \frac{(v_i - N\pi_i)^2}{N\pi_i}, \tag{46}$$

kde jsou π_i jsou teoretické pravděpodobnosti vypočtené podle vztahů

$$\begin{aligned} P(T = 0) &= \frac{1}{2}, \\ P(T = k) &= \frac{1}{2^{2k}}, \quad \text{pro } k = 1, 2, .., \\ P(T = k) &= \frac{1}{2^{2|k|+1}}, \quad \text{pro } k = -1, -2, .., \end{aligned}$$

p-hodnota se získá jako hodnota funkce

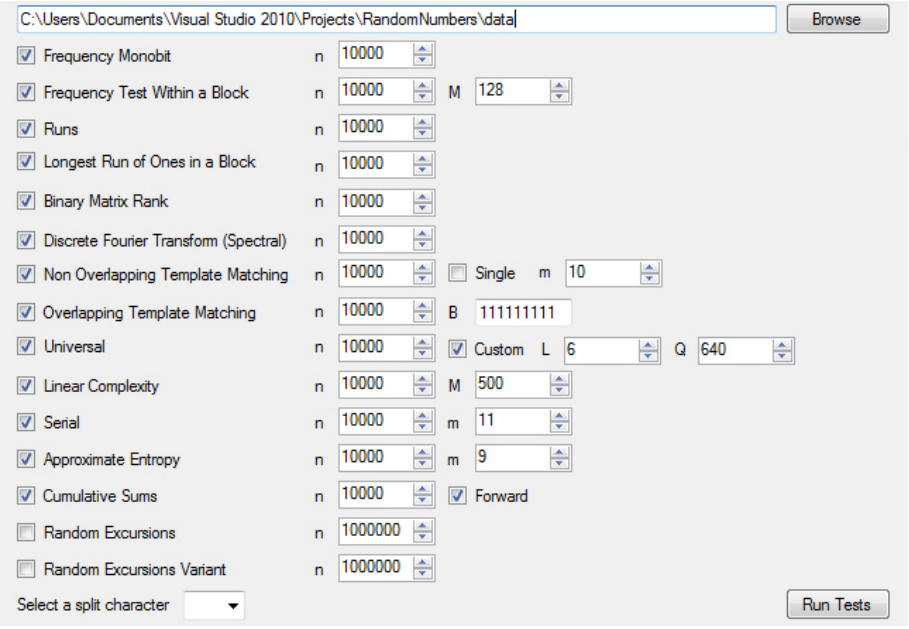
$$\text{igamc}\left(\frac{6}{2}, \frac{\chi^2(obs)}{2}\right). \tag{47}$$

6 Testování generátorů

V této kapitole se zabývám testováním generátorů náhodných čísel implementovaných ve vybraných softwarech. Testované aplikace jsou Microsoft Excel, MATLAB, Maple, programovací jazyk C a programovací jazyk Java.

Vstupními daty pro testovací baterii NIST je sekvence bitů délky n. Použité generátory generují čísla v desítkové soustavě, a tak je nutné vygenerované hodnoty převést do soustavy dvojkové. Testované sekvence byly získány převodem čísel z intervalu $\langle 0, 1023 \rangle$. Pro zachování vlastností vygenerovaných dat a zachování korektního rozdělení nul a jedniček, byla všechna čísla zapsána pomoci deseti bitů, tedy např. číslo 4 nebylo zapsáno v „klasickém“ tvaru 100_2, ale ve tvaru 0000000100_2. Testované sekvence byly vytvořeny z 1000 vygenerovaných hodnot, tudíž jejich délka je $n = 10000$.Každý test byl pětkrát opakován.

Výstupem testů je p-hodnota. p-hodnota vyjadřuje, jaká je minimální hladina významnosti α, na níž bychom mohli hraničně zamítnout nulovou hypotézu. Testovací baterie pracuje s hladinou významnosti $\alpha = 0,01$. Pokud p-hodnota nabude hodnoty vyšší než α, nulová hypotéza H_0 není zamítnuza a software vypíše success. Nulová hypotéza je pro všechny testy stejná a tvrdí, že testovaná sekvence pochází z uniformního rozdělení.



Obrázek 5: Ukázka prostředí testovací baterie

6.1 Programovací jazyk C 4.9.9.2

Programovací jazyk C vznikl v 70. letech minulého století. Patří mezi nejpopulárnější programovací jazyky. Nejčastěji se používá pro psaní systémového softwaru, ale je rozšířený i při tvorbě aplikací. Programovací jazyk C je díky velkému množství knihoven mocným nástrojem, schopným vytvářet ovladače, jádro operačního systému nebo aplikace využívající složitější matematiku. Generování náhodných čísel v C je umožněno pomocí LCG 3.1.

Název testu	P-hodnota	P-hodnota	P-hodnota	P-hodnota	P-hodnota
Frekvenční test	0,52217	0,98404	0,50925	0,73386	0,95216
Blokový frekvenční test	0,74139	0,56519	0,64485	0,55729	0,69719
Test sérií	0,59673	0,35757	0,41468	0,45998	0,58922
Blokový test nejdelší série	0,21197	0,23723	0,40004	0,29172	0,32228
Hodnost binární matice	0,37430	0,49951	0,86246	0,71385	0,15749
Spektrální test	0,52063	0,71357	0,85438	0,63288	0,85438
Shoda nepřek.se řetězců	0,49295	0,56501	0,47905	0,50742	0,51011
Shoda přek. se řetězců	0,57992	0,82218	0,73159	0,59485	0,62681
Univerzální statistický test	0,50727	0,98567	0,62963	0,78294	0,90668
Test lineární složitosti	0,80871	0,78322	0,77022	0,75656	0,95942
Sériový test	0,87868	0,52385	0,65684	0,55341	0,69029
Test přibližné entropie	0,68512	0,54261	0,55206	0,72709	0,63653
Test narůstajících součtů	0,91846	0,78759	0,61503	0,69421	0,98149

Tabulka 1: Výsledky testů programovacího jazyka C

6.2 Microsoft Excel 2003

Microsoft Excel je tabulkový editor, který je součástí kancelářského balíku Microsoft Office. Excel nabízí k dispozici přes 300 funkcí, výpočetní a grafické nástroje, kontigenční tabulky a programování pomocí maker. Od roku 1993 je Excel nejrozšířenější aplikací ve své oblasti. Starší verze Excelu používaly PRNG, který byl znám svou špatnou kvalitou. Od roku 2003 využívá Excel ke generování PRNG s názvem AS 183 [4], který kombinuje několik LCG 3.1.

Název testu	P-hodnota	P-hodnota	P-hodnota	P-hodnota	P-hodnota
Frekvenční test	0,65216	0,35757	0,37886	0,72034	0,33706
Blokový frekvenční test	0,57626	0,80455	0,52182	0,66491	0,33681
Test sérií	0,98407	0,37261	0,32103	0,77941	0,69709
Blokový test nejdelší série	0,70264	0,52162	0,43287	0,78477	0,33481
Hodnost binární matice	0,86246	0,94954	0,94954	0,33069	0,49951
Spektrální test	0,64636	0,85438	0,53288	0,31277	0,92688
Shoda nepřek. se řetězců	0,53749	0,53046	0,43946	0,49515	0,51869
Shoda přek. se řetězců	0,37101	0,32227	0,33341	0,64938	0,75158
Univerzální statistický test	0,80055	0,45377	0,64211	0,38293	0,50353
Test lineární složitosti	0,51829	0,43436	0,79606	0,74382	0,74382
Sériový test	0,33927	0,65482	0,55985	0,67439	0,31179
Test přibližné entropie	0,70689	0,44458	0,47152	0,52828	0,86875
Test narůstajících součtů	0,97841	0,63839	0,45964	0,87426	0,19383

Tabulka 2: Výsledky testů Excel

6.3 Programovací jazyk Java SE 8

Java je objektově orientovaný jazyk, který v roce 1995 vyvinula firma Sun Microsystems. Po jazyku C je Java považována za druhý nejrozšířenější programovací jazyk. Java je interpretovaný jazyk, místo skutečného strojového kódu vytváří tzv. mezikód, který není závislý na architektuře daného zařízení. Tento mezikód je pak interpetován pomocí tzv. virtuálního stroje Javy, který je k dispozici pro téměř všechny počítače a zařízení. Od roku 2007 je Java vyvíjena jako open source, což napomohlo jejímu rozšíření. Nevýhodou Javy oproti C je její rychlost, jelikož použití virtuálního stroje je obvykle časově náročné. Jako PRNG Java využívá LCG 3.1.

Název testu	P-hodnota	P-hodnota	P-hodnota	P-hodnota	P-hodnota
Frekvenční test	0,90448	0,96809	0,60306	0,52217	0,96809
Blokový frekvenční test	0,81729	0,62513	0,47373	0,38855	0,70275
Test sérií	0,88855	0,45929	0,77740	0,47403	0,61709
Blokový test nejdelší série	0,96001	0,96805	0,53811	0,87085	0,48857
Hodnost binární matice	0,58701	0,94954	0,86246	0,58701	0,71385
Spektrální test	0,71357	0,16867	0,16867	0,92688	0,27081
Shoda nepřek. se řetězců	0,52860	0,53644	0,54677	0,57047	0,49645
Shoda přek. se řetězců	0,73516	0,30313	0,12770	0,41697	0,50790
Univerzální statistický test	0,80585	0,48087	0,43821	0,71370	0,55548
Test lineární složitosti	0,90031	0,56957	0,97160	0,50581	0,54249
Test přibližné entropie	0,38732	0,37248	0,85836	0,28999	0,66540
Sériový test	0,72259	0,89355	0,50582	0,57446	0,45563
Test narůstajících součtů	0,32297	0,61103	0,94859	0,64761	0,93133

Tabulka 3: Výsledky testů programovacího jazyka Java

6.4 Maple 17

Maple je komerční počítačový algebraický systém vyvíjený společností Maplesoft. Uživateli nabízí provádění symbolických i numerických výpočtů, vytváření grafů, hypertextových zápisníků a programů. Maple používá vlastní programovací jazyk podobný Pascalu, který obsahuje předdefinované funkce a procedury. Tyto funkce pokrývají mnoho odvětví matematiky od základů diferenciálního a integrálního počtu, statistiky, lineární algebry, až k řešení diferenciálních rovnic a logice. Generování náhodných čísel v Maplu je prováděno pomoci MT 3.4 generátoru.

Název testu	P-hodnota	P-hodnota	P-hodnota	P-hodnota	P-hodnota
Frekvenční test	0,95216	0,19360	0,32709	0,73386	0,46151
Blokový frekvenční test	0,67842	0,54905	0,86658	0,65544	0,59468
Test sérií	0,81036	0,54637	0,80226	0,16185	0,77974
Blokový test nejdelší série	0,11776	0,84658	0,49299	0,17605	0,47879
Hodnost binární matice	0,33069	0,37431	0,86246	0,15749	0,94954
Spektrální test	0,85438	0,58191	0,64636	0,19889	0,92688
Shoda nepřek. se řetězců	0,48370	0,49688	0,52030	0,54062	0,53467
Shoda přek. se řetězců	0,43372	0,81077	0,53902	0,37993	0,65751
Univerzální statistický test	0,99214	0,10093	0,59657	0,43821	0,57243
Test lineární složitosti	0,89003	0,41196	0,48133	0,90031	0,19729
Test přibližné entropie	0,40963	0,25751	0,41159	0,84454	0,72621
Sériový test	0,54997	0,56019	0,30898	0,48719	0,83529
Test narůstajících součtů	0,80579	0,22367	0,35393	0,75052	0,23751

Tabulka 4: Výsledky testů Maple

6.5 MATLAB R2013b

MATLAB je vysokoúrovňový programovací jazyk a interaktivní výpočetní prostředí vyvíjené firmou MathWorks. MATLAB umožňuje vykreslování grafů, analýzu dat, práci s maticemi, implementaci algoritmů, vytváření uživatelského prostředí a spolupráci s externími programy napsaných v jiných jazycích. Původně byl MATLAB určen hlavně pro inženýrsko-matematické účely, ale časem byl rozšířen a dnes se používá v široké paletě aplikací. MATLAB je hojně využíván jak ve vědeckotechnických institucích, tak v průmyslových společnostech. Ke generování náhodných čísel využívá MATLAB MT 3.4 generátor.

Název testu	P-hodnota	P-hodnota	P-hodnota	P-hodnota	P-hodnota
Frekvenční test	0,53526	0,90448	0,41222	0,77948	0,57548
Blokový frekvenční test	0,48968	0,27350	0,19341	0,24242	0,60729
Test sérií	0,82287	0,33713	0,64067	0,74956	0,13279
Blokový test nejdelší série	0,94428	0,68028	0,15491	0,99531	0,40751
Hodnost binární matice	0,37431	0,15904	0,15749	0,58701	0,64839
Spektrální test	0,92688	0,14203	0,35880	0,64636	0,40886
Shoda nepřek. se řetězců	0,47427	0,48897	0,48334	0,50992	0,52048
Shoda přek. se řetězců	0,90564	0,88447	0,22521	0,74228	0,32768
Univerzální statistický test	0,27181	0,41293	0,22724	0,55801	0,44424
Test lineární složitosti	0,77022	0,31160	0,84532	0,69006	0,34938
Test přibližné entropie	0,74900	0,64002	0,33861	0,42016	0,22329
Sériový test	0,49326	0,19045	0,41970	0,11067	0,39856
Test narůstajících součtů	0,39388	0,61103	0,69421	0,89733	0,95864

Tabulka 5: Výsledky testů MATLAB

6.6 Zhodnocení výsledků

Na základě získáných p-hodnot z pěti měření nezamítáme nulovou hypotézu u všech provedených testů a testovaných generátorů. Zjištěné p-hodnoty nabyly hodnot vyšších než $0,01$, nemůžeme tedy vyvrátit hypotézu, že se jedná o náhodně vygenerované vzorky. Rozdílné p-hodnoty stejných testů u jednotlivých generátorů jsou zapříčiněny volbou počatečních vstupů. I když jsou některé p-hodnoty nižší než jiné, na samotné nezamítnutí nulové hypotézy to nemá žádný vliv. Test se dá považovat za nespolehlivý, pokud p-hodnota nabyde hodnoty z intervalu $(0.01,\ 0.05)$, v tomto případě se doporučuje provést nové měření. Na základě získaných p-hodnot lze testované generátory LCG a MT implementované v daných aplikacích ohnodnit jako přiměřeně kvalitní.

7 Závěr

Hlavním cílem práce je seznámení s metodami generování pseudonáhodných čísel a způsoby ověřování jejich kvality. V teoretické části byly uvedeny vybrané pasáže ze statistiky a teorie pravděpodobnosti. Dále byly popsány nejznámnější deterministické algoritmy určené ke generování pseudonáhodných čísel s uniformním rozdělením a metody vytváření náhodné veličiny s jiným pravděpodobnostním rozdělením. Druhá část práce se zabývala statistickými testy z testovací baterie, které byly následně aplikovány na několik softwarových generátorů. Tyto generátory statistickými testy prošly, což svědčí o jejich dostatečné kvalitě.

Při opakování testů bylo nutné se vypořádat s odlišnými tvary formátu výstupu u jednotlivých programů. Vzhledem k tomu, že testovací baterie byla určena pro posloupnosti bitů, bylo nutné korektně zvolit vhodný způsob zápisu.

8 Reference

[1] Coddington, Paul D. Random Number Generators for Parallel Computers, Syracuse University, 1997

[2] Kopf, Tomáš. Aplikovaná statistika, Slezská univerzita v Opavě, 2013

[3] Rukhin, Andrew a kol. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST, 2010

[4] Wichmann, B.A, Hill, I.D. Algorithm AS 183: An Efficient and Portable Psuedo-Random Number Generator, Middlesex, 2005.

[5] Random Number Generators [online]. 30. 5. 2001. Dostupné z `https://www.cs.indiana.edu/~kapadia/project2/node6.html` [citováno 9. 4. 2014].

[6] Xiang Tian, Khaled Benkrid. Mersenne Twister Random Number Generation on FPGA, CPU and GPU [online]. The University of Edinburgh. `http://www.see.ed.ac.uk/~SLIg/papers/tian_AHS09.pdf` [citováno 17. 4. 2014]

[7] Erin Casey. Berlekamp-Massey Algorithm [online]. University of Minnesota, 2000. Dostupné z `http://www.math.umn.edu/~garrett/students/reu/MB_algorithm.pdf` [citováno 1. 5. 2014]

[8] Litschmannová, Martina. Vybrané kapitoly z pravděpodobnosti [online]. VŠB – TU Ostrava, Fakulta elektrotechniky a informatiky, 2011. Dostupné z `mi21.vsb.cz` [citováno 4. 5. 2014]

[9] Litschmannová, Martina. Úvod do statistiky [online]. VŠB – TU Ostrava, Fakulta elektrotechniky a informatiky, 2011. Dostupné z `mi21.vsb.cz` [citováno 4. 5. 2014]

[10] Antoch, J. Jak pomocí simulace dokázat nemožné, Informační Bulletin České Statistické Společnosti, ročník 9., č. 1, 1998

[11] Máša, P. Zajímavý generátor náhodných čísel, Informační Bulletin České Statistické Společnosti, ročník 14., č. 4, 2003

A Tabulky

A.1 Příloha k blokovému testu nejdelší série

Třídy	Teoretické pravděpodobnosti
$\{ v \leq 1 \}$	$\pi_0 = 0,2148$
$\{ v = 2 \}$	$\pi_1 = 0,3672$
$\{ v = 3 \}$	$\pi_2 = 0,2305$
$\{ v \geq 4 \}$	$\pi_3 = 0,1872$

Tabulka 6: Rozdělení do tříd a jejich teoretické pravděpodobnosti pro $K = 3$, $M = 8$

Třídy	Teoretické pravděpodobnosti
$\{ v \leq 4 \}$	$\pi_0 = 0,1174$
$\{ v = 5 \}$	$\pi_1 = 0,2430$
$\{ v = 6 \}$	$\pi_2 = 0,2493$
$\{ v = 7 \}$	$\pi_3 = 0,1752$
$\{ v = 8 \}$	$\pi_4 = 0,1027$
$\{ v \geq 9 \}$	$\pi_5 = 0,1124$

Tabulka 7: Rozdělení do tříd a jejich teoretické pravděpodobnosti pro $K = 5$, $M = 128$

Třídy	Teoretické pravděpodobnosti
$\{ v \leq 6 \}$	$\pi_0 = 0,1174$
$\{ v = 7 \}$	$\pi_1 = 0,2460$
$\{ v = 8 \}$	$\pi_2 = 0,2523$
$\{ v = 9 \}$	$\pi_3 = 0,1755$
$\{ v = 10 \}$	$\pi_4 = 0,1027$
$\{ v \geq 11 \}$	$\pi_5 = 0,1124$

Tabulka 8: Rozdělení do tříd a jejich teoretické pravděpodobnosti pro $K = 5$, $M = 512$

Třídy	Teoretické pravděpodobnosti
$\{ v \leq 7 \}$	$\pi_0 = 0,1307$
$\{ v = 8 \}$	$\pi_1 = 0,2437$
$\{ v = 9 \}$	$\pi_2 = 0,2452$
$\{ v = 10 \}$	$\pi_3 = 0,1714$
$\{ v = 11 \}$	$\pi_4 = 0,1002$
$\{ v \geq 12 \}$	$\pi_5 = 0,1088$

Tabulka 9: Rozdělení do tříd a jejich teoretické pravděpodobnosti pro $K = 5$, $M = 1000$

Třídy	Teoretické pravděpodobnosti
$\{ v \leq 10 \}$	$\pi_0 = 0,0882$
$\{ v = 11 \}$	$\pi_1 = 0,2092$
$\{ v = 12 \}$	$\pi_2 = 0,2483$
$\{ v = 13 \}$	$\pi_3 = 0,1933$
$\{ v = 14 \}$	$\pi_4 = 0,1208$
$\{ v = 15 \}$	$\pi_5 = 0,0675$
$\{ v \geq 16 \}$	$\pi_6 = 0,0727$

Tabulka 10: Rozdělení do tříd a jejich teoretické pravděpodobnosti pro $K = 6$, $M = 10000$

A.2 Příloha k testu lineární složitosti

$T_i \leq -2,5$	Inkrementace v_0 o jeden
$-2,5 < T_i \leq -1,5$	Inkrementace v_1 o jeden
$-1,5 < T_i \leq -0,5$	Inkrementace v_2 o jeden
$-0,5 < T_i \leq 0,5$	Inkrementace v_3 o jeden
$0,5 < T_i \leq 1,5$	Inkrementace v_4 o jeden
$1,5 < T_i \leq 2,5$	Inkrementace v_5 o jeden
$T_i > 2,5$	Inkrementace v_6 o jeden

Tabulka 11: Pravidla inkrementace tříd